NASA TM X-1541

A FORTRAN PROGRAM FOR COMPUTATION OF MASS IN EARTH ORBIT REQUIRED FOR INTERPLANETARY MISSIONS

By Samuel W. Pitts and Alfred C. Mascy

NASA Headquarters
Mission Analysis Division
Moffett Field, Calif.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

A FORTRAN PROGRAM FOR COMPUTATION OF MASS IN EARTH ORBIT REQUIRED FOR INTERPLANETARY MISSIONS

By Samuel W. Pitts and Alfred C. Mascy

NASA Headquarters
Moffett Field, Calif.

SUMMARY

A necessary parameter to be investigated in the analysis of space missions is the total assembled mass in Earth orbit, as it is a strong indicator in comparisons of the performance of various propulsion systems. The computer code described in this paper was designed to compute the mass in Earth orbit for round-trip stopover missions, flyby missions, and orbiter missions with a single-stage high-thrust system for each propulsive phase. The actual gravity losses suffered during each propulsive phase are computed by complete integration of the equations of motion, thereby permitting the propulsion system to be optimized. The program allows the input of planet ephemeris data so that it may be used for any planet. The following options are available to the user.

The number of engines may be specified or optimized for each propulsive phase of the mission.

Planet capture may be performed by aerobraking or propulsive maneuvers.

A specific number of engines/tanks may be specified.

The engine burn time may be constrained to a desired value.

Midcourse correction penalty may be specified.

Time dependent life support may be included for manned missions.

The inert weight fractions may be specified as constants or computed by the program.

An engine clustering penalty may be specified.

Arrival or departure or both from elliptical orbits may be specified.

Thrust may be initiated or terminated at points other than periapsis.

Perturbing of the parking orbit may be investigated, i.e., by using an apoapsis kick.

Each phase of a mission may be run independently.

Thrust vector orientation other than tangential may be specified.

Examples provided include the complete input and output for a round-trip mission, a flyby mission, and an orbiter mission.

INTRODUCTION

The analytical computation of the assembled mass in Earth orbit for space missions is time consuming and difficult. Approximations are used in such computations for the gravity losses suffered during propulsive phases, the inert weight scaling laws, etc. Therefore, a computer code employing numerical integration of the equations of motion, optimization procedures on several variables, and iterative procedures for inert calculations has been developed that enables the user to compute accurately the mass in Earth orbit for round-trip missions, orbiter missions, and flyby missions.

The flyby type mission consists of one single propulsive phase, Earth departure, and is assumed to place a specified weight on a heliocentric flyby trajectory. The orbiter type mission consists of two phases, Earth departure and planet capture. Both phases may be propulsive or the capture phase may be performed by an aerobraking maneuver. A midcourse correction fuel penalty may be provided.

The round-trip stopover mission has three phases, Earth departure, planet capture, and planet departure, and includes a provision for an excursion module for landing on the target planet. The excursion module weight must include the propulsion necessary to descend to and ascend from the planet and any staged weight left at the planet (i.e., small probes). The Earth return weight would include the mission module Earth entry vehicle and time dependent expendables (e.g., life support). Any propulsion fuel consumed in midcourse can be included. This computer code has been written in Fortran IV and has been used on a 7094 computer. The fact that Fortran IV was used as a program language should make it applicable to other types of computing equipment.

Required program input, output, example problems and the program listings are included here as appendixes A-D, respectively.

ANALYSIS

Method of Solution

The procedures for computing total assembled mass in Earth orbit for round trip stopover missions, orbiter missions, and flyby missions are outlined in this paper. The program performs the calculations for the round trip missions in three distinct phases (i.e., phase I, planet departure, phase 2, planet capture, and phase 3, Earth departure), starting with the Earth return weight and working in reverse order through the total missions. The orbiter mission consists only of phases 2 and 3 and a flyby mission only of phase 3. The inputs that the user must provide include propulsion system characteristics (e.g., engine weight, engine thrust), trajectory data (e.g., V_∞, orbit altitudes), mission module weights, staged weights at planet and time dependent expendables (e.g., life support). A complete listing of all required and optional input is included in appendix A.

The actual gravity losses suffered during each propulsive phase of a mission are computed by numerical integration of the equation of motion in order to determine the optimum ratio of thrust to weight of each stage, the inert fractions, propellant consumption, and the engine burn times. The programs also have an option that allows the mass in Earth orbit to be computed by the closed form ballistic equations rather than integration of the equations of motion, and includes an approximation for the gravity losses in a separate subroutine. Other available options are described in the section Program Options.

This computer code is capable of computing four separate types of missions. The mission type must be specified by the input parameter MODE as follows:

MODE = 1	Round trip stopover missions
MODE = 2	Orbiter missions
MODE = 3	Flyby missions
MODE = 4	Only the planet capture phase is computed

The method of solution described below is for a round trip stopover mission with a specified Earth return module which contains an Earth entry vehicle. Starting with the Earth return module the program works through the mission profile in reverse order in the following manner. The planet departure payload is set equal to the weight of the Earth return module plus the midcourse fuel and time dependent expendables. An initial estimate of the ratio of total engine thrust to total vehicle weight is computed by the program for the planet departure phase. This initial estimate is computed from the closed-form ballistic equation with no gravity losses. The trajectory is then integrated using the estimated T/W_1 until the specified hyperbolic excess velocity for planet departure is obtained, at which time the propulsive factor P is computed.

$$P = \frac{1.0 + (WE/WL)}{MFST\ (1 + A) - A}$$

where

WE engine weight

WL Earth return weight + time dependent expendables + midcourse fuel

A inert fraction (either input or allowed to be computed)

MFST propellant consumed during powered phase

and

$$MFST = 1.0 - \frac{T/W * t}{Isp}$$

T/W engine thrust/gross weight of vehicle

t burn time, sec

Isp specific impulse, sec

This propulsion factor is then multiplied by the returned payload weight to arrive at a total gross weight of the stage based on the previous integration. Forming the T/W_2 based on the new found gross weight of the stage, the program compares the initial estimate for T/W_1 to the T/W_2 computed to see whether the two values agree within a desired tolerance. If the values disagree by more than the desired tolerance, the trajectory is rerun with T/W_2 and the iteration continues until convergence is obtained. The value of the gross weight, WG = P * WL, is then taken as the gross weight of the planet departure stage. The ratio of propellant weight to gross weight WP/WG is given by WP/WG = 1.0 - MFST and the weight of propellant is computed from WP = WP/WG * WG. The weight of the inerts (less the engines) is found from WT = WP * A where A is the inert fraction computed from the scaling law equations or is an input constant.

The payload weight of the capture phase of the mission is set equal to the gross weight of the planet departure phase plus the weight of any payload left at the planet.

$$WL1 = WG + WEM$$

where WEM is the weight of staged payload (e.g., planet excursion module). An initial weight of propellant is assumed by the program and the initial estimate for the WTG, inert weight, is made, using the inert fraction computed from the scaling laws. Using these initial estimates, an estimate of the total gross weight of this stage less the propellant is given by

$$WGL = WL1 + WE + WTG$$

where

WE weight of engines

WTG initial estimate of inert weight (less engine)

WL1 gross weight of departure stage + staged payload.

The value of WG2 is used to form the thrust-to-weight ratio to start the integration procedure. The equations of motion are integrated using negative time until the desired hyperbolic excess velocity, V_∞, is obtained for the capture phase. The use of -t in the integration allows the equation

$$MFST = 1.0 - (T/W * t/Isp)$$

to add propellant to the vehicle as the trajectory is computed in reverse order starting from the low capture orbit. When the desired V_∞ is obtained the propulsive factor P is computed

$$P = \frac{1 + (WE/WL1)}{(1.0/MFST)\ (1.0 + A) - A}$$

The product P * WL1 then gives the total weight, WBO, of the capture stage prior to the start of the capture maneuver. The total weight of propellant used during the capture phase is

$$WFC = WBO - WG2$$

and the weight of the inerts (less the engine), WTC, is

$$WTC = WPC * A$$

where A is the inert fraction. This value of WTC is compared to the initial guess WTG to see whether the values are within the desired tolerance. If the tolerance is met, the gross weight of the capture phase of the mission is the WBO found. If the tolerance is not met, WTC is used as the next guess of WTG and the iteration continues until convergence is obtained.

When the third phase of the mission, Earth departure, is initiated, the gross weight of the capture phase of the mission is used as a base weight and is increased as follows. The payload of the third phase is

$$WLE = WBO + WM + WLS$$

where

WBO gross weight of capture phase

WM weight of midcourse fuel

WLS weight of time dependent expendables (e.g., life support)

This value of WLE is then used to form the thrust/weight needed to start the integration of the equation of motion for the Earth departure phase. The iteration procedure described in the planet departure phase of the mission is the same employed for the Earth departure and all values of inert weights, propellant weights and gross weights are found by the same method.

Program Options

There are several options in this computer code that are available to the user. They are listed here and explained further in the section devoted to the computer input.

(1) A limit on engine operating time has been set in the program to 1800 sec. This may be changed by reading in TLIMIT equal to the desired operating time limit in seconds. When this limit is exceeded, the number of engines used is increased by one until the operating time constraint is met.

(2) The number of engines used per propulsive phase may be fixed or optimized at the discretion of the user. When a fixed number of engines is to be used (no optimization), set SEARCH = 0 and NØE1, NØE2, NØE3 equal to desired number of engines. (Here and in the following 1, 2, and 3 refer to planet departure, planet capture, and Earth departure maneuvers, respectively.) The computer code will not change the number of engines specified unless the operating time limit is exceeded; therefore, set TLIMIT to a large positive number if no operating limit is desired. The user may set SEARCH = 1 and the engine optimization procedure is followed; that is, the number of engines specified by NØE1, NØE2, NØE3 is used as a starting base and the number of engines is reduced by one until a minimum gross weight is obtained. The optimum number of engines is then output with the associated gross weight. The operating time limit will also be a factor and should be set to the desired limit when optimizing the number of engines.

(3) Either a specific engine thrust level may be used as noted in (2) above, or the engine may be "rubberized" as follows: Estimate a total thrust level needed for the phase of the mission being computed. Divide this thrust level by the value of NØE1, NØE2, or NØE3 pertinent to the mission phase and set ET1, ET2, or ET3 equal to the thrust thus obtained. Also, set SEARCH = 1, allowing the program to optimize the number of engines. If the output lists the number of engines as 1, choose a lower thrust per engine and rerun the problem. If the number of engines is equal to the input value, choose a higher thrust per engine and repeat the run. When the output number of engines is greater than 1 and less than the input value, the number of "rubberized" engines has been optimized.

(4) Aerobraking for the capture phase of a mission may be used by setting IAERØ = 2 and assigning a heat shield penalty to the incoming vehicle weight. If a penalty of 15 percent of the vehicle weight was desired, the user would input AERØP = 1.15.

(5) The inert fraction (less the engine) may be an input or may be computed in the program from the following equations which were developed by United Aircraft Corp. under contract NAS 2-2928. Tankage structure weight WS:

$$WS = \frac{A^{*} TN^{0.1}}{\sigma^{0.533}} WP^{0.9} + K_1 \, TN^{0.1}$$

where

A	constant
σ	specific gravity of propellant
K_1	constant for fixed weights
WP	propellant weight
TN	number of tanks

The number of tanks is set initially at 1.0. If the user specifies the number of engines per tank, NET1, then the program will determine the optimum number of tanks as well as the optimum number of engines. If a value for NET1, NET2, or NET3 is not read as input, the program will compute the mission using one single tank for each phase that has not specified a particular number of engines per tank.

(6) Boil off and thermoinsulation for cryogenic tanks

$$WBO + W_i = B \left(\frac{WP}{\sigma}\right)^{2/3} \left(\frac{t\Delta T}{L}\right)^{1/2}$$

where

B	constant in kms system
WP	weight of usable propellant
σ	specific gravity of propellants
t	time of exposure, days
L	latent heat of vaporization, kms system
ΔT	temperature difference across insulation thickness

(7) To avoid integration of micrometeoroid impacts with respect to exposure time in estimating weight of meteoroid protection, and at the same time include the effects of the relatively large fluxes of debris in the asteroid belt, the following approximate equation has been used:

$$WM = C\left(\frac{WP}{\sigma}\right)^{5/6} t^{1/4}$$

The choice of the values assigned to the constant, C, should distinguish between missions to regions of the solar system inside and within or beyond the asteroid belt. The following values of C are suggested.

	Parameter C	
	0 to 1.8 A.U.	1.8 A.U. & beyond
High	0.06	0.10
Nominal	.03	.05
Low	.015	.025

These values of C are for single-sheet thicknesses, with time t measured in hours. Inasmuch as the tank structure WS affords some protection against meteoroid penetration, only an additional weight, ΔWM, need to be added to the tank structure. If the calculated meteoroid protection, WM, is less than the tank structure, WS, no additional weight is added to the tank. If WM is greater than WS but less than 1.33 WS, additional weight, ΔWM = WM - WS, is added to the single tank sheet. If the required meteoroid protection is greater than 1.33 WS, a second tank sheet is used to construct a Whipple bumper with the additional weight ΔWM = WS/3.

(8) For arrival at or departure from elliptical orbits the periapsis and apoapsis radii (RPP and RAP, respectively) are specified. If the true anomaly is not specified, the program will initiate thrust at the periapsis. To terminate or initiate thrust at points along the orbit other than periapsis TLT1, TLT2, and TLT3 must be specified equal to the desired true anomaly.

(9) The user may also perturb the parking orbits by such maneuvers as an apogee kick by placing KICK1, KICK2, or KICK3 equal to the change in periapsis radius desired at the respective orbit.

(10) If fixed inert fractions rather than calculated inerts are desired, simply place ACØMP = 1 and input the desired inert fractions A1, A2, and A3.

(11) A fixed Earth departure stage, such as a Saturn SII, may be treated by placing AT3 = 0.0 and WK3 = 37500.0 which is approximately the fixed tank inerts on an SII stage. With ACØMP = 2, this same procedure may be used at any phase of the mission.

(12) A useful option at the user's command is the ability to operate the program at two different levels of accuracy. With LEVEL = 1, all propulsive phases of the mission are computed in closed form with an estimate of the

gravity losses, which is an empirical curve fit to the results of reference 1. With LEVEL = 2, all propulsive phases are computed with a complete integration of the equations of motions. The first level is useful for scanning many missions because of the minimal machine time. The second level, although somewhat longer in computer time, allows a more exacting analysis of a mission. LEVEL = 2 is built into the code and will perform the integrations if it is not changed by the input LEVEL = 1 as noted above.

(13) A clustering penalty of 10 percent per additional engine has been included in the program but may be changed by input. If for example a 12-percent penalty were to be used, set PEN = 0.12 in the mission data input.

General Comments

Stacked cases.- Cases may be stacked by simply placing an asterisk card between each set of input. Only those input quantities which are to be changed need be added after the asterisk. If no input is read for the inert calculation, two (2) asterisk cards must follow the stacked mission data.

Perturbed inputs.- The program will utilize the last input card encountered for a given parameter in its analysis. Hence, if a case requires the changing of one or more input cards from a prior case, one can simply place the desired changes on a card at the end of the pertinent input and it will utilize these latter values in the analysis. Thus, whole problem decks can be duplicated for stacked cases and perturbations made using supplementary cards without having to repunch the original problem deck.

All the variables in the Input Parameter List, appendix A, have been given mode numbers in the left-hand column, i.e., 1, 2, 3, 4. These refer to the type mission requiring that particular variable. The four modes are as follows:

MØDE = 1   Round-trip mission (all variables marked 1)
MØDE = 2   Orbiter mission (all variables marked 2)
MØDE = 3   Flyby mission (all variables marked 3)
MØDE = 4   The capture phase only is computed (all variables marked 4).

It should be noted that some of the variables are needed for all four mission modes while some are only necessary for one particular mission mode. Mode 4 may be used for investigation of propulsion system characteristics for the capture phase only. If a survey of only the departure phase is desired, the user may set MODE = 3, and GSE, RGE, RPE and RAE equal to the departure planet's respective values.

National Aeronautics and Space Administration
Moffett Field, Calif., 94035, Nov. 7, 1967
789-40-01-08-00-15

REFERENCE

1. Long, R. S.: Escape from a Circular Orbit with Finite Velocity at Infinity. Astronautica Acta, Sept. 1958.

APPENDIX A

INPUT PARAMETERS

Mode	Variable name	Description
1	EW1	Weight/engine, lb, planet depart
1, 2, 4	EW2	Weight/engine, lb, planet capture
1, 2, 3	EW3	Weight/engine, lb, Earth depart
1	ET1	Thrust/engine, lb, planet depart
1, 2, 4	ET2	Thrust/engine, lb, planet capture
1, 2, 3	ET3	Thrust/engine, lb, Earth depart
1	NØE1	Number of engines, planet depart
1, 2, 4	NØE2	Number of engines, planet capture
1, 2, 3	NØE3	Number of engines, Earth depart
1	I1	Isp, sec, planet depart
1, 2, 4	I2	Isp, sec, planet capture
1, 2, 3	I3	Isp, sec, Earth depart
1	A1	Inert fraction (less engine), planet depart
1, 2, 4	A2	Inert fraction (less engine), planet capture
1, 2, 3	A3	Inert fraction (less engine), Earth depart
1	V8CØ1	V_∞, km/sec, planet depart
1, 2, 4	V8CØ2	V_∞, km/sec, planet capture
1, 2, 3	V8CØ3	V_∞, km/sec, Earth depart
1	MCPF1	1.XX - midcourse correction penalty, inbound leg
1, 2, 3	MCPF	1.XX - midcourse correction penalty, outbound leg
1, 2, 3, 4	WPL	Weight of MM + EEM, etc., lb
1	WEM	Weight of MEM, lb
1, 2, 3, 4	GSE	"g" at Earth, km/sec^2
1, 2, 3, 4	RAE	Apogee of departure orbit at Earth, ⊕ radii
1, 2, 3, 4	RPE	Perigee of departure orbit at Earth, ⊕ radii
1, 2, 3, 4	RGE	Radius of Earth, km
1, 2, 4	RAP	Apogee of capture orbit at planet, planet radii
1, 2, 4	RPP	Perigee of capture orbit at planet, planet radii
1, 2, 4	RGP	Radius of destination planet, km
1, 2, 4	GSP	"g" at destination planet, km/sec^2
1	FELLØW	Number of men
1	LEGT1	Inbound leg time, days
1	LEGT2	Outbound leg time, days
1	LSD	Life support requirements, lb/man-day
1, 2, 4	AERØP	1.XX - penalty for aerobraking at destination planet
1, 2, 3, 4	SEARCH	0 - will not optimize number of engines but use input values 1 - will optimize number of engines

1, 2, 3, 4	ACØMP	1 - will use fixed inerts 2 - will calculate inerts using inert subroutine
1, 2, 4	IAERØ	1 - propulsive capture at destination planet 2 - aerobraking at destination planet
1, 2, 3, 4	PEN	Engine clustering penalty
1, 2, 3, 4	TLIMIT	Maximum allowable engine burn time, sec
1, 2, 3, 4	NET1	Number of engines/tank, planet departure, XX.X
1, 2, 4	NET2	Number of engines/tank, planet capture, XX.X
1, 2, 3	NET3	Number of engines/tank, Earth departure, XX.X
1, 2, 3, 4	MØDE	The type mission to be computed
1, 2, 3, 4	LEVEL	1 - propulsive phases computed in closed form with approximation for gravity loss 2 - propulsive phases computed by numerical integration of equation of motion
1	KICK1	Desired % change in periapsis radius, planet depart
1, 2, 4	KICK2	Desired % change in periapsis radius, planet capture
1, 2, 3	KICK3	Desired % change in periapsis radius, Earth depart
1	TLT1	True anomaly at initiation of thrust, planet depart, deg
1, 2, 4	TLT2	True anomaly at initiation of thrust, planet capture, deg
1, 2, 3	TLT3	True anomaly at initiation of thrust, Earth depart, deg
1, 2, 3, 4	*	An asterisk in any column (1-72) will terminate the reading of the above mission data.

The following input quantities are used in the inert calculations. This data must follow the above mission data. If ACØMP = 1 in input above, the constant inert fractions A1, A2, A3 above are used, and the following input may be deleted.

1	SIG1	Propellant specific gravity, planet depart
1, 2, 4	SIG2	Propellant specific gravity, planet capture
1, 2, 3	SIG3	Propellant specific gravity, Earth depart
1	AT1	Constant (A) in tankage equation, planet depart
1, 2, 4	AT2	Constant (A) in tankage equation, planet capture
1, 2, 3	AT3	Constant (A) in tankage equation, Earth depart
1	BØ1	Constant (B) in boil off equation, planet depart
1, 2, 4	BØ2	Constant (B) in boil off equation, planet capture
1, 2, 3	BØ3	Constant (B) in boil off equation, Earth depart

1	CM1	Constant (C) in meteoroid equation, planet depart
1, 2, 4	CM2	Constant (C) in meteoroid equation, planet capture
1, 2, 3	CM3	Constant (C) in meteoroid equation, Earth depart
1	TEMP1	ΔT in boil-off equation, $^{\circ}K$, planet depart
1, 2, 4	TEMP2	ΔT in boil-off equation, $^{\circ}K$, planet capture
1, 2, 3	TEMP3	ΔT in boil-off equation, $^{\circ}K$, Earth depart
1	HEAT1	Propellant latent heat of vaporization (L) in boil-off equation, kcal/kg, planet depart
1, 2, 4	HEAT2	Propellant latent heat of vaporization (L) in boil-off equation, kcal/kg, planet capture
1, 2, 3	HEAT3	Propellant latent heat of vaporization (L) in boil-off equation, kcal/kg, Earth depart
1	TEX1	Exposure time (t) boil-off and meteoroid equation, days, planet depart
1, 2, 4	TEX2	Exposure time (t) boil-off and meteoroid equation, days, planet capture
1, 2, 3	TEX3	Exposure time (t) boil-off and meteoroid equation, days, Earth depart
1	WK1	Constant K in tankage equation, kg, planet depart
1, 2, 4	WK2	Constant K in tankage equation, kg, planet capture
1, 2, 3	WK3	Constant K in tankage equation, kg, Earth depart
1, 2, 3, 4	*	An asterisk in any column (1-72) will terminate the reading of the inert data and execution will begin

APPENDIX B

OUTPUT PARAMETERS

The following output will appear for each phase of the mission computed.

Variable name	Description
Engines	Optimum number of engines
Thrust	Total thrust, lb
Engine weight	Total engine weight, lb
VINF	V_∞, km/sec
VGL	Gravity losses, km/sec
DEL V	ΔV including gravity losses, km/sec
Burn time	Stage burn time, sec
Isp	Stage Isp, sec
A	Inert fraction (exclusive of engine wt) $\equiv W_I/W_P$
MUL	Payload fraction $\equiv W_L/W_O$
Tanks	Number of tanks for that stage
Payload	Total stage payload, lb
W TANK	Weight of all inerts except engine $(W_T + W_{BO} + W_I + W_{MM})$, lb
W FUEL	Weight of propellant used, lb
GRØSS WT	Gross weight, lb
P	Propulsion factor $\equiv 1/\mu L$
T/W	Stage initial thrust to weight
MID Fuel	Midcourse correction penalty, lb
LIFE WT	Life support weight, lb
THETA	Angle between original line of apsides and departure asymptote
EX.MØD	Excursion module weight, lb
Tank	W_T/W_P
Boil off and insulation	$(W_{BO} + W_I)/W_P$
Meteoroid shield	W_{MM}/W_P

APPENDIX C

EXAMPLE PROBLEMS

Included in this section are sample runs of a manned roundtrip, an unmanned orbiter, an unmanned flyby, and a capture phase only. It is noted that an asterisk card should appear between the mission input data and the system inert parameter data.

In the first example, nuclear propulsion was used at all phases; however, the thrust level of an individual engine was different. Planet departure and capture was performed with a small 75,000 lb thrust engine while the Earth departure phase used a 200,000 lb thrust engine. The input SEARCH = 1 was used allowing the number of engines to be optimized. The operating time limit is set to 1,000 sec, and the number of engines per tank NET1 = NET2 = NET3 = 2.0.

The program increased the number of engines to two for the planet departure phase and to three for the planet capture phase because fewer engines exceeded the operating time constraint.

The optimization indicates three engines, for the Earth departure phase produced the minimum total gross weight while remaining within the operating time limit. The separate inert fraction for tankage, boil off and insulation, and meteoroid shielding are output for the user's reference. Note that the boil off and insulation fraction are zero for the Earth departure phase because TEX3 = 0.0 (the exposure time) was assumed.

Example II is for a Mercury orbiter mission. Note that the planet parameters RAP, RPP, RGP, and GSP have been changed to those for Mercury. The parameter from the input list with a mode 2 designation only have been supplied. The engine operating time limit was not specified and the 1800 sec built in was used. This example will have one tank only since the number of engines per tank NET2, NET3 was not read in. The output consists only of two phases printed because no planet departure phase was computed. The input value MØDE = 2 was required to accomplish this orbiter mission.

Example III is a flyby to Mercury. The value MØDE = 3 is read in and only those input parameters for mode 3 are needed. The output as noted will consist of only an Earth departure phase. The inert fractions printed show a value for the tankage while the boil off and insulation and meteoroid shielding are zero. The boil off and insulation are zero as TEX3 (exposure time) was set to 0.0 and the meteoroid shielding is zero, indicating the tankage will afford the needed meteoroid shielding based on the scaling law. TEX3 may be input as the assembly time in Earth orbit for manned mission or the parking orbit time for unmanned missions.

```
                 EXAMPLE  OF  INPUT  FOR  ROUND-TRIP  MISSION  (MODE=1)
VTIT=60H  SAMPLE MARS STOPOVER MISSION                                  ,
MODE=1,
EW1=17000.0,  ET1=75000.0,  NOE1=1,
EW2=17000.0,  ET2=75000.0,  NOE2=1,
EW3=28700.0,  ET3=200000.0,  NOE3=6,
I1=850.0, I2=850.0, I3=850.0,
 A1=.25,  A2=.25, A3=.25,
V8C01=6.92,  V8C02=3.59,  V8C03=2.865,
MCPF=1.02,  MCPF1=1.02,
WPL=92000.0,  WEM=134000.0,
RAE=1.066,  RPE=1.066,
RAP=1.265,  RGP=3410.0,  RPP=1.265,  GSP=.00369,
SEARCH=1,  ACOMP=2,
FELLOW=8.0,  LEGT1=260.0,  LEGT2=180.0,  LSD=3.75,
PEN=.1,
NET1=2.0,  NET2=2.0,  NET3=2.0,
TLIMIT=1000.0,
*
INPUT FOR INERT ROUTINE
SIG1=.077,  SIG2=.077,  SIG3=.077,
AT1=.10,  AT2=.10,  AT3=.10,
B01=.034,  B02=.034,  B03=.034,
CM1=.06,  CM2=.06,  CM3=.06,
TEMP1=156.0,  TEMP2=156.0,  TEMP3=156.0,
HEAT1=108.1,  HEAT2=108.1,  HEAT3=108.1,
TEX1=210.0,  TEX2=180.0,  TEX3=0.0,
WK1=500.0,  WK2=500.0,  WK3=500.0,
*
```

SAMPLE MARS STOPOVER MISSION

DEPARTURE FROM PLANET

ENGINES	THRUST	ENG. WT	VINF	VGL	DEL.V	BURN TIME	ISP	A	MUL	TANKS
2	150000.0	37400.0	6.920	0.0712	5.1502	855.75	850.0	0.262	0.3059	1.0

PAY LOAD	W TANK	W FUEL	GROSS WT	P	T/W	MID FUEL	LIFE WT	THETAP
101640.0	40061.2	153119.0	332220.2	3.269	0.4515	1840.0	7800.0	114.6

INERT FRACTION A CONSISTS OF TANK=0.136192 BOIL OFF + INSULATION=0.080044 METEROID SHIELD=0.04540

CAPTURE PHASE

ENGINES	THRUST	ENG. WT	VINF	VGL	DEL.V	BURN TIME	ISP	A	MUL	TANKS
3	225000.0	56100.0	3.590	0.0265	2.5982	796.66	850.0	0.258	0.5918	2.0

PAY LOAD	W TANK	W FUEL	GROSS WT	P	T/W	EX MOD	THETAP
466220.2	54540.2	211172.9	787763.9	1.690	0.2856	134000.0	-257.9

INERT FRACTION A CONSISTS OF TANK=0.144018 BOIL OFF + INSULATION=0.066249 METEROID SHIELD=0.04801

DEPARTURE FROM EARTH

ENGINES	THRUST	ENG. WT	VINF	VGL	DEL.V	BURN TIME	ISP	A	MUL	TANKS
3	600000.0	94710.0	2.865	0.0558	3.6006	736.33	850.0	0.126	0.5415	2.0

PAY LOAD	W TANK	W FUEL	GROSS WT	P	T/W	MID FUEL	LIFE WT	THETAP
808919.2	66260.7	523990.0	1493879.9	1.847	0.4016	15755.3	5400.0	174.4

INERT FRACTION A CONSISTS OF TANK=0.126454 BOIL OFF + INSULATION=0.000000 METEROID SHIELD=0.00000

EXAMPLE OF INPUT FOR ORBITER MISSION (MODE=2)

```
VTIT=60H  SAMPLE MERCURY ORBITER MISSION                            ,
MODE=2,
EW2=312.0,  ET2=25000.0,  NOE2=6,
EW3=28700.0,  ET3=200000.0,  NOE3=1,
I2=450.0,  I3=850.0,
V8CO2=8.04,  V8CO3=11.47,
MCPF=1.02,
WPL=10000.0,
RAE=1.036,  RPE=1.036,
RAP=1.041,  RPP=1.041,  RGP=2420.0,  GSP=.00372,
SEARCH=1,  ACOMP=2,
*
INPUT FOR INERT SUBROUTINE
SIG2=.077,  SIG3=.077,
AT2=.10,  AT3=.10,
BO2=.034,  BO3=.034,
CM2=.06,  CM3=.06,
TEMP2=156.0,  TEMP3=156.0,
HEAT2=108.1,  HEAT3=108.1,
TEX2=130.0,  TEX3=0.0,
WK2=500.0,  WK3=500.0,
SIG2=.33,
TEMP2=104.0,
HEAT2=60.5,
*
```

SAMPLE MERCURY ORBITER MISSION

CAPTURE PHASE

ENGINES	THRUST	ENG. WT	VINF	VGL	DEL.V	BURN TIME	ISP	A	MUL	TANKS
2	50000.0	686.4	8.040	0.0614	6.1726	487.98	450.0	0.131	0.1385	1.0

PAY LOAD	W TANK	W FUEL	GROSS WT	P	T/W	EX MOD	THETAP
10000.0	7141.7	54404.0	72180.8	7.218	0.6927	0.0	-271.3

INERT FRACTION A CONSISTS OF TANK=0.086059 BOIL OFF + INSULATION=0.036615 METEROID SHIELD=0.00860

DEPARTURE FROM EARTH

ENGINES	THRUST	ENG. WT	VINF	VGL	DEL.V	BURN TIME	ISP	A	MUL	TANKS
1	200000.0	28700.0	11.470	0.3081	8.4223	981.14	850.0	0.128	0.2036	1.0

PAY LOAD	W TANK	W FUEL	GROSS WT	P	T/W	MID FUEL	LIFE WT	THETAP
74010.6	29652.0	231194.2	363556.8	4.912	0.5501	1451.2	0.0	138.5

INERT FRACTION A CONSISTS OF TANK=0.128256 BOIL OFF + INSULATION=0.000000 METEROID SHIELD=0.00000

EXAMPLE OF INPUT FOR FLY-BY MISSION (MODE=3)

```
VTIT=60H  SAMPLE MERCURY FLYBY MISSION                                  ,
MODE=3,
EW3=17000.0,  ET3=75000.0,  NOE3=1,
I3=850.0,
V8CO3=11.47,
MCPF=1.02,
WPL=10000.0,
RAE=1.036,  RPE=1.036,
RAP=1.041,  RPP=1.041,  RGP=2420.0,  GSP=.00372,
SEARCH=1,  ACOMP=2,
*
INPUT FOR INERT SUBROUTINE (P002IN)
SIG3=.077,
AT3=.10,
BO3=.034,
CM3=.06,
TEMP3=156.0,
HEAT3=108.1,
TEX3=0.0,
WK3=500.0,
*
```

SAMPLE MERCURY FLYBY MISSION

DEPARTURE FROM EARTH

ENGINES	THRUST	ENG. WT	VINF	VGL	DEL.V	BURN TIME	ISP	A	MUL	TANKS
1	75000.0	17000.0	11.470	0.1839	8.2981	720.58	850.0	0.158	0.1013	1.0

PAY LOAD	W TANK	W FUEL	GROSS WT	P	T/W	MID FUEL	LIFE WT	THETAP
10200.0	10020.7	63500.7	100721.3	9.875	0.7446	200.0	0.0	131.2

INERT FRACTION A CONSISTS OF TANK=0.157804 BOIL OFF + INSULATION=0.000000 METEROID SHIELD=0.00000

```
                    EXAMPLE  OF  INPUT  FOR  CAPTURE  PHASE  (MODE=4)
VTIT=60H  EXAMPLE CAPTURE PHASE ONLY                                   ,
MODE=4,
EW2=312.0,  ET2=25000.0,  NOE2=6,
I2=450.0,
V8CO2=8.04,
WPL=10000.0,
RAE=1.036,  RPE=1.036,
RAP=1.041,  RPP=1.041,  RGP=2420.0,  GSP=.00372,
SEARCH=1,  ACOMP=2,
*
INPUT FOR INERT SUBROUTINE (P007IN)
AT2=.10,
BO2=.034,
CM2=.06,
TEX2=130.0,
WK2=500.0,
SIG2=.33,
TEMP2=104.0,
HEAT2=60.5,
*
```

EXAMPLE CAPTURE PHASE ONLY

CAPTURE PHASE

ENGINES	THRUST	ENG. WT	VINF	VGL	DEL.V	BURN TIME	ISP	A	MUL	TANKS
2	50000.0	686.4	8.040	0.0614	6.1726	487.98	450.0	0.131	0.1385	1.0

PAY LOAD	W TANK	W FUEL	GROSS WT	P	T/W	EX MOD	THETAP
10000.0	7141.7	54404.0	72180.8	7.218	0.6927	0.0	-271.3

INERT FRACTION A CONSISTS OF TANK=0.086059 BOIL OFF + INSULATION=0.036615 METEROID SHIELD=0.00860

```
C      MAIN CONTROL PROGRAM FOR REQUIRED MASS IN ORBIT COMPUTATION
       COMMON ALTO,PSI1,I12,GE,GM,ALT,T(70),PSI2,K,GS
       COMMON/BOXGL/    DVI,FW12,V8CO,VSU,DVGL,LEVEL,TST,RG,RLTS,AS
       COMMON/BOX1/A1,A2,A3,A12,IABC,ACOMP,SIG1,SIG2,SIG3,AT1,BO1,CM1,AT2
      1,BO2,CM2,AT3,BO3,CM3,TEMP1,TEMP2,TEMP3,HEAT1,HEAT2,HEAT3,TEX1,TEX2
      2,TEX3,WK1,WK2,WK3,TN
       DIMENSION VTIT(18),VIN8(20),MF8(10),THOL8(10),TVIN8(10),TMF8(10),T
      1THOL8(10),T99(10),TT99(10)
       REAL I11,I12,MF12,ML10,ML11,ML12,ML1,K,KICK,MUL,MFST,MF8,I1,I2,I3,
      1MCPF,MCFW,KICK1,KICK2,KICK3,MCPF1,MCFW1,LEGT1,LEGT2,LSD,NET1,NET2,
      2NET3,NET
       INTEGER SEARCH,ACOMP,OKNOE
       EXTERNAL DIVE
       MCPF=1.0
       MCPF1=1.0
       ACOMP=1
       LEVEL=2
       RAE=1.04355
       RPE=1.04355
       RGE=6375.445
       GSE=.980665E-2
       PSI=0.0
       SEARCH=0
       KICK1=1.0
       KICK2=1.0
       KICK3=1.0
       TLT1=0.0
       TLT2=0.0
       TLT3=0.0
       IAERO=1
       NOE1=1
       NOE2=1
       NOE3=1
       MODE=1
       NET1=-1.0
       NET2=-1.0
       NET3=-1.0
       TN=1.0
       PEN=.1
       TLIMIT=1800.0
       FELLOW=0.0
       LEGT1=0.0
       LEGT2=0.0
       LSD=0.0
  645  TC=3600.0
       GE=.980665E-2

       VB=0.0
       DEG=1.745329E-2
       PY=180.0*DEG
       TERROR=1.0E-6
       IABC=1
       DTBACK=2.0
       TIMEI=0.0
       DELTI=4.0
       K=1.0
       WGMIN=1.0F+10
       IOK1=2
       IOK2=2
       IOK3=2
    1  CALL INPUT(2HI1,I1,2HI2,I2,2HI3,I3,2HA1,A1,2HA2,A2,2HA3,A3,2HB1,B1
      1,4HNET1,NET1,4HNET2,NET2,4HNET3,NET3,3
      2HEW1,EW1,3HEW2,EW2,3HEW3,EW3,3HET1,ET1,3HET2,ET2,3HET3,ET3,3HWEM,W
      3EM,3HWPL,WPL,5HV8CO1,V8CO1,5HV8CO2,V8CO2,5HV8CO3,V8CO3,4HNOE1,NOE1
      4,4HNOE2,NOE2,4HNOE3,NOE3,3HGSP,GSP,3HRGP,RGP,3HRPP,RPP,3HRAP,RAP,5
      5HIAERO,IAERO,4HVTIT,VTIT,4HFW12,FW12,4HMCPF,MCPF,3HRAE,RAE,3HRPE,R
      6PE,3HRGE,RGE,3HGSE,GSE,5HTLTSI,TLTSI,3HPSI,PSI,5HDELTI,DELTI,5HAER
      7OP,AEROP,5HKICK1,KICK1,5HKICK2,KICK2,5HKICK3,KICK3,4HTLT1,TLT1,4HT
      8LT2,TLT2,4HTLT3,TLT3,6HSEARCH,SEARCH,5HMCPF1,MCPF1,5HACOMP,ACOMP,6
      9HFELLOW,FELLOW,3HLSD,LSD,5HLEGT1,LEGT1,5HLEGT2,LEGT2,4HMODE,MODE,3
      1HPEN,PEN,6HTLIMIT,TLIMIT,5HLEVEL,LEVEL)
       GO TO (1112,1111),ACOMP
 1111 CALL INPUT(4HSIG1,SIG1,4HSIG2,SIG2,4HSIG3,SIG3,3HAT1,AT1,3HAT2,AT2
      1,3HAT3,AT3,3HBO1,BO1,3HBO2,BO2,3HBO3,BO3,3HCM1,CM1,3HCM2,CM2,3HCM3
      2,CM3,5HTEMP1,TEMP1,5HTEMP2,TEMP2,5HTEMP3,TEMP3,5HHEAT1,HEAT1,5HHEA
      3T2,HEAT2,5HHEAT3,HEAT3,4HTEX1,TEX1,4HTEX2,TEX2,4HTEX3,TEX3,3HWK1,W
      4K1,3HWK2,WK2,3HWK3,WK3)
 1112 WRITE(6,100)(VTIT(I),I=1,10)
  100  FORMAT(1H1,20X,10A6)
       IF(SEARCH.LE.0) GO TO 350
       IF(NOE1.GT.1) IOK1=1
       IF(NOE2.GT.1) IOK2=1
       IF(NOE3.GT.1) IOK3=1
  350  WL=WPL
       MCFW1=(WL*MCPF1)-WL
       WL=WL*MCPF1
       WLSE1=LEGT1*FELLOW*LSD
       WLSE2=LEGT2*FELLOW*LSD
       WL=WL+WLSE1
       WP=WL
       NOE=NOE1
       WE=EW1*FLOAT(NOE)
       THRUST=ET1*FLOAT(NOE)
```

```
      ET=ET1
      EW=EW1
      V8CO=V8CO1
      A12=A1
      I12=I1
      KICK=KICK1
      NET=NET1
      TLTSI=TLT1
      GS=GSP
      RG=RGP
      RP=RPP
      RA=RAP
      DELT=DELTI
      PSI2=PSI*DEG
      FW12=THRUST/(WL+WP+WE)
      DVGL=0.0
      GO TO (9999,3333,4444,3333),MODE
 3333 WG=WPL
      WEM=0.0
      GO TO 1113
 4444 IAERO=2
      AEROP=1.0
      WEM=0.0
      WG=WPL
      GO TO 1113
 9999 IF(NET.LE.0.0)GO TO 25
      TN=FLOAT(NOE)/NET
      TN=TN+.9
      IOTN=TN
      TN=IOTN
      GO TO 9911
   25 TN=1.0
 9911 CALL INERT(WP,TANK,BOIS,SFMM)
      IF(NOE.EQ.1) GO TO 27
      WE=FLOAT(NOE)*PEN*EW+FLOAT(NOE)*EW
      THRUST=FLOAT(NOE)*ET
      GO TO 21
   27 WE=EW
      THRUST=ET
      GO TO 21
   21 GO TO(221,66,221),IABC
  221 WEWL=WE/WL
      TLTSR=TLTSI*DEG
      GM=GS*RG*RG
      VC=SQRT(GM/(RP*RG))
      AS=.5*(KICK*RP+RA)

      EPS=(RA-KICK*RP)/(RA+KICK*RP)
      PS=AS*RG*(1.0-EPS*EPS)
      RLTS=PS/(1.0+EPS*COS(TLTSR))
      RDLTS=SQRT(GM/PS)*EPS*SIN(TLTSR)
      VSU=SQRT((GM/(AS*RG))*(1.+2.*EPS*COS(TLTSR)+EPS*EPS)/(1.-EPS*EPS))
      DVI=SQRT((V8CO**2)+(2.0*GM/(RLTS    )))-VSU
   87 CALL INERT(WP,TANK,BOIS,SFMM)
      DVIWGL=DVI+DVGL
      BMFST = EXP((-DVIWGL)/(I12*GE))
      BPROP = (1.0+WEWL)/(BMFST*(1.0+A12)-A12)
      FW12 = THRUST/(BPROP*WL)
      WP2 = (1.0-BMFST) * (BPROP*WL)
      IF(ABS(1.0-(WP/WP2))-0.01) 88,88,89
   89 WP=WP2
      GO TO 87
   88 MFST = BMFST
      FW12=THRUST/(BPROP*WL)
    2 TIME=TIMEI
   75 ALTO=FW12*GE
      XX3=GM/(PS**3)
      THDT=SQRT(XX3   )*((1.0+EPS*COS(TLTSR))**2)+(VB/RLTS)
      GO TO (166,566), LEVEL
  166 CALL GLOSS
      TST=WP*I12/THRUST
      DVIWGL=DVI+DVGL
      MFST=EXP((-DVIWGL)/(I12*GE))
      IF(IABC.EQ.2) MFST=1.0/MFST
      GO TO 60
C     FOLLOWING IS SETUP FOR INTEGRATION ROUTINE
  566 T(2)=TIME
      T(3)=DELT
      T(4)=0.0
      T(5)=RLTS
      T(6)=THDT
      T(7)=RDLTS
      T(8)=0.0
      CALL INT(T,5,0,TERROR,0,0,0   ,0,0,DIVE)
   55 CALL INTM
  533 VSQ=T(7)*T(7)+((T(5)*T(6))**2)
      V=SQRT(VSQ)
      CAPE=.5*VSQ-GM/T(5)
      H=T(5)*T(5)*T(6)
      P=H*H/GM
      E=SQRT(1.0+(2.0*CAPE*P/GM))
      GAMMA=ATAN(T(7)/(T(5)*T(6)))
      TOP=T(7)/(T(5)*T(6))
```

```
      BOT=1.0-(T(5)/P)
      THETA=ARTN(TOP,BOT)
      IF(THETA.LT.0.0) THETA=THETA+360.0*DEG
      MF12=1.0-( (FW12*T(2))/(I12    ))
      FWBO=FW12/MF12
      IF(CAPE)10,11,11
   10 VINF=0.0
      GO TO 50
   11 VIF2=2.0*CAPE
      VINF=SQRT(VIF2)
      IF(VINF-V8CO)50,50,51
   50 HOLD2=T(2)
      HOLD4=T(4)
      HOLD5=T(5)
      HOLD6=T(6)
      HOLD7=T(7)
      HOLD8=T(8)
      GO TO 55
   51 T(2)=HOLD2
      T(3)=DTBACK
      T(4)=HOLD4
      T(5)=HOLD5
      T(6)=HOLD6
      T(7)=HOLD7
      T(8)=HOLD8
      KS=1
      CALL INT(T,5,1,0,0,0,0,0,0,DIVE)
   53 CALL INTM
      VSQ=T(7)*T(7)+((T(5)*T(6))**2)
      V=SQRT(VSQ)
      CAPE=.5*VSQ-GM/T(5)
      IF(CAPE.LT.0.0) GO TO 53
      H=T(5)*T(5)*T(6)
      P=H*H/GM
      E=SQRT(1.0+(2.0*CAPE*P/GM))
      GAMMA=ATAN(T(7)/(T(5)*T(6)))
      TOP=T(7)/(T(5)*T(6))
      BOT=1.0-(T(5)/P)
      THETA=ARTN(TOP,BOT)
      IF(THETA.LT.0.0) THETA=THETA+360.0*DEG
      REF=THETA-T(4)-TLTSR
      ASS=90.0*DEG+ASIN(1.0/E)
      TOUT7=(ASS-REF)/DEG
      MF12=1.0-( (FW12*T(2))/(I12    ))
      VIF2=2.0*CAPE
      VINF=SQRT(VIF2)

      IF(KS-5)54,54,56
   54 VIN8(KS)=VINF
       MF8(KS)=MF12
       THOL8(KS)=T(2)
      T99(KS)=TOUT7
      KS=KS+1
      GO TO 53
   56 DO 57 IJ=1,4
       TVIN8(IJ)=VIN8(IJ+1)
       TMF8(IJ)=MF8(IJ+1)
       TTHOL8(IJ)=THOL8(IJ+1)
      TT99(IJ)=T99(IJ+1)
   57 CONTINUE
       TVIN8(5)=VINF
       TMF8(5)=MF12
       TTHOL8(5)=T(2)
      TT99(5)=TOUT7
       DO 58 I4=1,5
       VIN8(I4)=TVIN8(I4)
      MF8(I4)=TMF8(I4)
       THOL8(I4)=TTHOL8(I4)
      T99(I4)=TT99(I4)
   58 CONTINUE
       DNN4=0.0
       IF(VINF-V8CO)53,53,59
   59 CALL TAINT(VIN8,MF8,V8CO,MFST,5,2,NIR,DNN4,THOL8,TST,T99,TTTOT)
   60 GO TO(61,62,61),IABC
   61 PROP=(1.0+WEWL)/(MFST*(1.0+A12)-A12)
      WIGUS=PROP*WL
      FWTST=THRUST/WIGUS
      IF(ABS(FW12-FWTST)-.01)70,70,71
   71 FW12=FWTST
      WPWO=1.0-MFST
      WO=PROP*WL
      WP=WPWO*WO
      CALL  INERT(WP,TANK,BOIS,SFMM)
      GO TO 75
   70 GO TO(777,62,63),IABC
  777 WPWO=1.0-MFST
      WTWO=WPWO*A12
      MUL=1.0/PROP
      WO=PROP*WL
      WP=WPWO*WO
      WT=WTWO*WO
      WE=WEWL*WL
      VEMOS=V8CO/29.785
```

```
      DV=I12*GE*(-ALOG(MFST))
      TB=TST
      WG=WO
      VGL=DV-DVI
      ISAVE=1
      IF(TB.LT.TLIMIT) GO TO 22
      NOE=NOE+1
      IOK1=2
      GO TO 9999
   22 GO TO(110,666),IOK1
  666 ISAVE=2
      GO TO 176
  110 IF(WG-WGMIN)176,102,102
  101 IF(NOE.LE.1) GO TO 111
      NOE=NOE-1
      WGMIN=WG
      GO TO 9999
  102 WGMIN=1.0E+10
  111 WRITE(6,600)
  600 FORMAT(1H0,40X,21HDEPARTURE FROM PLANET)
      WRITE(6,601)
  601 FORMAT(1H0,3X,7HENGINES,3X,6HTHRUST,5X,7HENG. WT,4X,4HVINF,4X,5HVG
     1L  ,4X,5HDEL.V,4X,9HBURN TIME,4X,3HISP,7X,1HA,7X,3HMUL,3X,5HTANKS)
      WRITE(6,602)OKNOE,OKTHR,OKWE,V8CO,OKVGL,OKDV,OKTB,I12,OKA12,OKMUL,
     1OKTN
  602 FORMAT(7X,I2,5X,F8.1,3X,F8.1,3X,F6.3,2X,F6.4,3X,F6.4,3X,F8.2,5X,F6
     1.1,3X,F5.3,3X,F6.4,3X,F6.1)
      WRITE(6,603)
  603 FORMAT(1H0,3X,8HPAY LOAD,4X,6HW TANK,6X,6HW FUEL,3X,8HGROSS WT,7X,
     11HP,7X,3HT/W,5X,8HMID FUEL,5X,7HLIFE WT,3X,6HTHETAP)
      WRITE(6,604)OKWL,OKWT,OKWP,OKWG,OKP,OKFW,OKMF1,OKWLS,OKTT
  604 FORMAT(3X,F9.1,3X,F8.1,3X,F9.1,3X,F9.1,3X,F6.3,3X,F6.4,3X,F8.1,3X,
     1F7.1,3X,F8.1,3X,F6.2)
      IF(ACOMP.LE.1) GO TO 1113
      WRITE(6,605)OKTANK,OKBO,OKMM
  605 FORMAT(1H0,35HINERT FRACTION A CONSISTS OF  TANK=,F8.6,24H  BOIL O
     1FF + INSULATION=,F8.6,18H  METEROID SHIELD=,F7.5)
 1113 GO TO (700,800),IAERO
  700 DELT=-4.0
      WGMIN=1.0E+10
      PSI2=180.0*DEG
      V8CO=V8CO2
      NOE=NOE2
      ET=ET2
      EW=EW2
      WE=EW2*FLOAT(NOE)

      THRUST=ET2*FLOAT(NOE)
      NET=NET2
      A12=A2
      KICK=KICK2
      TLTSI=TLT2
      I12=I2
      IABC=2
      DTBACK=-2.0
      WL1=WG+WEM
      WL=WG
      WP=2.0E5
      CALLINERT(WP,TANK,BOIS,SFMM)
      WTGUS=WP*A12
      FW12=THRUST/(WL+WTGUS+WE+WEM)
      TLTSR=TLTSI*DEG
      GM=GS*RG*RG
      VC=SQRT(GM/(RP*RG))
      AS=.5*(KICK*RP+RA)
      EPS=(RA-KICK*RP)/(RA+KICK*RP)
      VP=SQRT((2.0*GM/RG)*((1.0/RP)-(1.0/(2.0*AS))))
      PS=AS*RG*(1.0-EPS*EPS)
      RDLTS=SQRT(GM/PS)*EPS*SIN(TLTSR)
      RLTS=PS/(1.0+EPS*COS(TLTSR))
      VSU=SQRT((GM/(AS*RG))*(1.+2.*EPS*COS(TLTSR)+EPS*EPS)/(1.-EPS*EPS))
      DVI=SQRT((V8CO**2)+(2.0*GM/(RLTS    )))-VSU
      DVGL=0.0
  143 CALL INERT(WP,TANK,BOIS,SFMM)
      WTGUS=WP*A12
      WE=EW2*FLOAT(NOE)
      THRUST=ET2*FLOAT(NOE)
      FW12=THRUST/(WL+WTGUS+WE+WEM)
      DVIWGL=DVI+DVGL
      BMFST=EXP((-DVIWGL)/(I12*GE))
      BPROP=(1.0+WE/WL1)/(BMFST*(1.0+A12)-A12)
      WP=(1.0-BMFST)*(BPROP*WL1)
      WTG1=WP*A12
      FW12=THRUST/(WL+WTG1+WE+WEM)
      IF(ABS(1.0-(WTGUS/WTG1))-.01)141,141,142
  142 WTGUS=WTG1
      GO TO 143
  141 WTGUS=A12*WP
      MFST=1.0/BMFST
      GO TO 9999
   66 WLC=WL+WTGUS+WE+WEM
      FW12=THRUST/WLC
      GO TO 2
```

```
 62  PROP=(1.0+WE/WL1)/((1.0/MFST)*(1.0+A12)-A12)
     WBO=PROP*WL1
     WPC=ABS(WBO-WLC)
     WTC=WPC*A12
     IF(ABS(1.0-WTC/WTGUS)-.01)65,65,67
 67  CALL INERT(WPC,TANK,BOIS,SFMM)
     WTGUS=WTC
     GO TO 66
 65  PROP=(1.0+WE/WL1)/((1.0/MFST)*(1.0+A12)-A12)
     MUL=1.0/PROP
     WO=PROP*WL1
     WP=WPC
     WT=WTC
     DV=I12*GE*(ALOG(MFST))
     VEMOS=V8CO/29.785
     WG=WO
     TB=ABS(TST)
     VGL=DV-DVI
     ISAVE=3
     IF(TB.LT.TLIMIT) GO TO 23
     NOE=NOE+1
     IOK2=2
     GO TO 9999
 23  GO TO(210,667),IOK2
667  ISAVE=4
     GO TO 176
210  IF(WG-WGMIN)176,202,202
201  WGMIN=WG
     IF (NOE.LE.1) GO TO 211
     NOE=NOE-1
     WTGUS=A12*(WL1-WEM)
     GO TO 9999
202  WGMIN=1.0E+10
211  WRITE(6,508)
     WGMIN=1.0E+10
508  FORMAT(1H0,40X,14HCAPTURE  PHASE)
     WRITE(6,601)
     WRITE(6,602)OKNOE,OKTHR,OKWE,V8CO,OKVGL,OKDV,OKTB,I12,OKA12,OKMUL,
    1OKTN
     WRITE(6,613)
613  FORMAT(1H0,3X,8HPAY LOAD,4X,6HW TANK,6X,6HW FUEL,3X,8HGROSS WT,7X,
    11HP,7X,3HT/W,5X,6HEX MOD,5X,6HTHETAP)
     WRITE(6,604)OKWC,OKWT,OKWP,OKWG,OKP,OKFW,WEM,OKTT
     IF(ACOMP.LE.1) GO TO 987
     WRITE(6,605)OKTANK,OKBO,OKMM
987  IF(MODE.EQ.4) GO TO 645

     IABC=3
     WRITE(6,503)
503  FORMAT(1H0,39X,20HDEPARTURE FROM EARTH)
     NOE=NOE3
     WE=EW3*FLOAT(NOE)
     THRUST=ET3*FLOAT(NOE)
     EW=EW3
     ET=ET3
     NET=NET3
     V8CO=V8CO3
     A12=A3
     KICK=KICK3
     TLTSI=TLT3
     I12=I3
     GS=GSE
     RG=RGE
     RP=RPE
     RA=RAE
     DELT=4.0
     PSI2=0.0
     DTBACK=2.0
     MCFW=(WG*MCPF)-WG
     WG=WG*MCPF
     WL=WG
     WL=WL+WLSE2
     WP=WL
     FW12=THRUST/(WL+WP+WE)
     DVGL=0.0
     GO TO 9999
800  IF(MODE.EQ.3) GO TO 987
     WRITE(6,500)
500  FORMAT(1H0,39X,27HCAPTURE PHASE AERO BREAKING)
     PLW=WG+WEM
     PLW2=AEROP*PLW
     HSW=PLW2-PLW
     WRITE(6,501)
501  FORMAT(1H0,3X,7HPAYLOAD,5X,6HEX.MOD,4X,8HHEAT SH.,4X,8HGROSS WT)
     WRITE(6,502)PLW,WEM,HSW,PLW2
502  FORMAT(3X,F9.1,3X,F8.1,3X,F8.1,3X,F9.1)
     WG=PLW2
     GO TO 987
 63  WPWO=1.0-MFST
     WTWO=WPWO*A12
     MUL=1.0/PROP
     WO=PROP*WL
     WP=WPWO*WO
```

```
      WT=WTWO*WO
      WE=WEWL*WL
      VEMOS=V8CO/29.785
      DV=I12*GE*(-ALOG(MFST))
      TB=TST
      WG=WO
      VGL=DV-DVI
      ISAVE=5
      IF(TB.LT.TLIMIT) GO TO 24
      NOE=NOE+1
      IOK3=2
      GO TO 9999
   24 GO TO(310,668),IOK3
  668 ISAVE=6
      GO TO 176
  310 IF(WG-WGMIN)176,302,302
  301 IF (NOE.LE.1) GO TO 311
      NOE=NOE-1
      WGMIN=WG
      GO TO 9999
  302 NOE=NOE+1
      IOK3=2
      WGMIN=1.0E+10
      GO TO 9999
  311 WRITE(6,601)
      WGMIN=1.0E+10
      WRITE(6,602)OKNOE,OKTHR,OKWE,V8CO,OKVGL,OKDV,OKTB,I12,OKA12,OKMUL,
     1OKTN
      WRITE(6,623)
  623 FORMAT(1H0,3X,8HPAY LOAD,4X,6HW TANK,6X,6HW FUEL,3X,8HGROSS WT,7X,
     11HP,7X,3HT/W,5X,8HMID FUEL,5X,7HLIFE WT,5X,6HTHETAP)
      WRITE(6,604)OKWL,OKWT,OKWP,OKWG,OKP,OKFW,OKMF2,OKWL2,OKTT
      IF(ACOMP.LE.1) GO TO 99
      WRITE(6,605)OKTANK,OKBO,OKMM
   99 GO TO 645
  176 OKNOE=NOE
      OKTHR=THRUST
      OKWE=WE
      OKVGL=VGL
      OKDV=DV
      OKTB=TB
      OKA12=A12
      OKMUL=MUL
      OKTN=TN
      OKWL=WL
      OKWC=WL1

      OKWT=WT
      OKWP=WP
      OKWG=WG
      OKP=PROP
      OKFW=THRUST/WG
      OKMF1=MCFW1
      OKWLS=WLSE1
      OKTT=TTTOT
      OKTANK=TANK
      OKBO=BOIS
      OKMM=SFMM
      OKMF2=MCFW
      OKWL2=WLSE2
      GO TO (101,111,201,211,301,311), ISAVE
      END
```

```
C       APPROXIMATE GRAVITY LOSS ROUTINE
      SUBROUTINE GLOSS
C
C     THIS ROUTINE IS AN APPROXIMATION FOR THE GRAVITY LOSSES, AND IS
C     USED ONLY FOR LEVEL=1. WHEN LEVEL=2 THE GRAVITY LOSSES ARE SET TO
C     0 AND RETURNED TO THE MAIN PROGRAM WHERE THE TRUE GRAVITY LOSSES
C     ARE FOUND BY INTEGRATION OF THE EQUATIONS OF MOTION. THIS ROUTINE
C     MAY BE REPLACED BY THE USER WITH HIS OWN APPROXIMATION IF DESIRED,
C     TO DO THIS ALL NEEDED VALUES MUST BE PLACED IN THE COMMON BLOCK
C     /BOXGL/ HERE AND IN THE MAIN PROGRAM THE VALUE OF THE GRAVITY
C     LOSSES MUST BE PLACED IN DVGL BEFORE RETURN TO THE MAIN ROUTINE
C
      COMMON ALTO,PSI1,I12,GE,GM,ALT,T(70),PSI2,K,GS
      COMMON/BOXGL/    DVI,FW12,V8CO,VSU,DVGL,LEVEL,TST,RG,RLTS,AS
      REAL I12
      GO TO (2,5), LEVEL
  2   DVG1=.036+.0693*((V8CO/VSU)**2)
      FACT1=1.0/(AS*AS)
      FACT2=(GS/GE)**2
      P1=FW12*.5
      P2=I12*GE
      P5=DVG1*DVG1*VSU*FACT1*FACT2/(P1*P1)
      P6=EXP(DVI/P2)
      G=DVGL
  9   P9=1.0+(P6*EXP(G/P2))
      PART1=2.0*P5*P6*EXP(G/P2)/P2
      PART2=P9**3
      FOFG=G-(P5/(P9*P9))
      FPRIM=1.0+(PART1/PART2)
      G1=G-(FOFG/FPRIM)
      IF(ABS(1.0-G1/G)-.001) 25,25,26
 26   G=G1
      GO TO 9
 25   DVGL=.5*(G1+G)
  5   RETURN
      END
```

```
C     THIS IS THE DERIVATIVE ROUTINE FOR THE INTEGRATION ROUTINE
      SUBROUTINE DIVE
      COMMON ALTO,PSI1,I12,GE,GM,ALT,T(70),PSI2,K,GS
      DEG=.01745329
      REAL K,I12
      TOP=T(5)*T(6)
      BOT=T(7)
      SIT=ARTN(TOP,BOT)
      IF(SIT.LT.0.0) SIT=SIT+360.0*DEG
      PSI1=PSI2+K*SIT
      PRT1=T(2)/(I12*GE)
      PRT2=1.0/ALTO
      ALT=1.0/(PRT2-PRT1)
      XX1=T(5)*T(6)*T(6)-(GM/(T(5)*T(5)))
      XX2=ALT*COS(PSI1)
      T(12)=XX1+XX2
      T(11)=(ALT*SIN(PSI1)/T(5))-(2.0*T(7)*T(6))/T(5)
      T(9)=T(6)
      T(10)=T(7)
      T(13)=0.0
      RETURN
      END
```

```
C     INERT SUBROUTINE
      SUBROUTINE INERT(WPI,TANK,BOIS,SFMM)
C
C     SUBROUTINE INERT COMPUTES THE INERT FRACTION FOR TANKAGE, BOIL-OFF
C     AND INSULATION, AND METEROID PROTECTION (INCLUDES WHIPPLE BUMPER)
C     EFFECT. ACOMP=1 INERT FRACTION IS INPUT CONSTANT ACOMP=2 INERT
C     FRACTION IS COMPUTED. IABC IS CONTROL FOR PHASE OF MISSION BEING
C     COMPUTED 1=PLANET DEPARTURE,2=PLANET CAPTURE,3=EARTH DEPARTURE
C
      COMMON/BOX1/A1,A2,A3,A12,IABC,ACOMP,SIG1,SIG2,SIG3,AT1,BO1,CM1,AT2
     1,BO2,CM2,AT3,BO3,CM3,TEMP1,TEMP2,TEMP3,HEAT1,HEAT2,HEAT3,TEX1,TEX2
     2,TEX3,WK1,WK2,WK3,TN
      INTEGER ACOMP
      WPI=WPI*.4535924
      GO TO (1,2),ACOMP
    1 GO TO (3,4,5),IABC
    3 A12=A1
      GO TO 90
    4 A12=A2
      GO TO 90
    5 A12=A3
      GO TO 90
    2 GO TO (7,8,9),IABC
    7 SIG=SIG1
      A=AT1
      B=BO1
      C=CM1
      TEMP=TEMP1
      EL=HEAT1
      EXTME=TEX1
      WK=WK1
      GO TO 80
    8 SIG=SIG2
      A=AT2
      B=BO2
      C=CM2
      TEMP=TEMP2
      EL=HEAT2
      EXTME=TEX2
      WK=WK2
      GO TO 80
    9 SIG=SIG3
      A=AT3
      B=BO3
      C=CM3
      TEMP=TEMP3

      EL=HEAT3
      EXTME=TEX3
      WK=WK3
   80 TAU=A/(SIG**.533)
   83 TANK=((TAU)*(TN**.1))/(WPI**.1)+(WK*TN)/WPI
  81  BOT=(WPI**.33333)*(SIG**.666667)
      BOIS=(B/BOT)*SQRT(EXTME*TEMP/EL)
      TOP=(EXTME**.25)/(SIG**(5.0/6.0))
      TOP1=1.0/(WPI**(1.0/6.0))
      SFMM=C*TOP*TOP1
      IF(SFMM-TANK)30,30,31
   30 SFMM=0.0
      GO TO 91
   31 WTEST=1.333333*TANK
      IF(SFMM-WTEST)32,32,33
   32 SFMM=SFMM-TANK
      GO TO 91
   33 WTEST2=(16.0/3.0)*TANK
      IF(SFMM-WTEST2)34,34,35
   34 SFMM=TANK/3.0
      GO TO 91
   35 SFMM=(SFMM/4.0)-TANK
   91 A12=TANK+BOIS+SFMM
      WWWP=WPI/.4535924
   90 WPI=WPI/.4535924
  110 RETURN
      END
```